Accelerating Molecular Modeling Applications with Graphics Processors

John Stone

Theoretical and Computational Biophysics Group

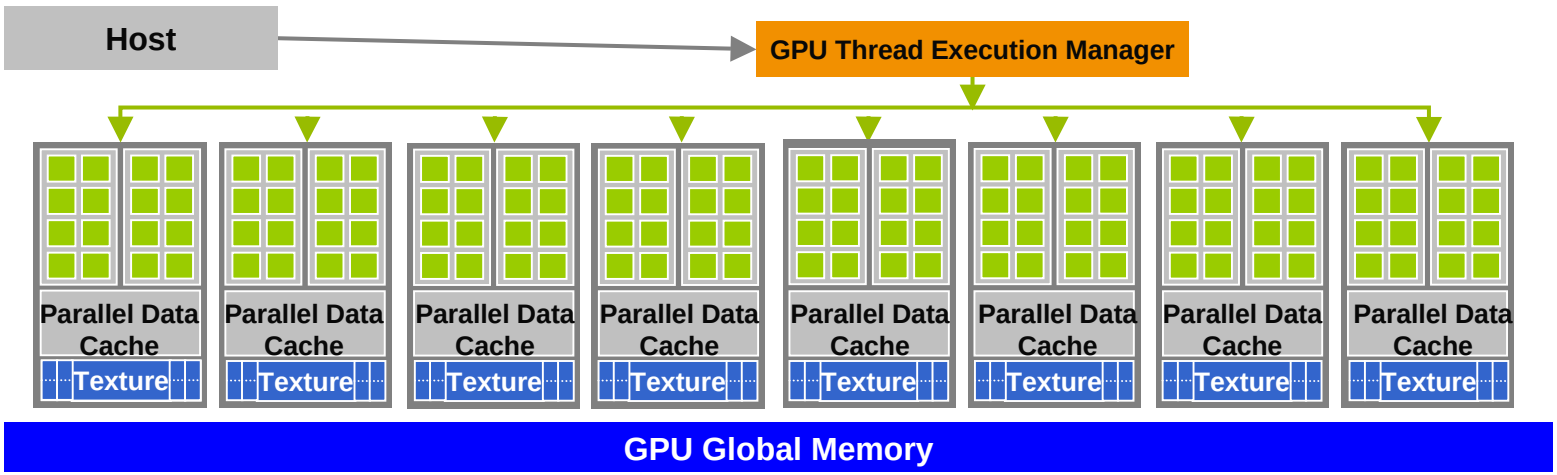
University of Illinois at Urbana-Champaign

http://www.ks.uiuc.edu/Research/gpu/

SIAM Conference on Parallel Processing for Scientific Computing, March 12, 2008

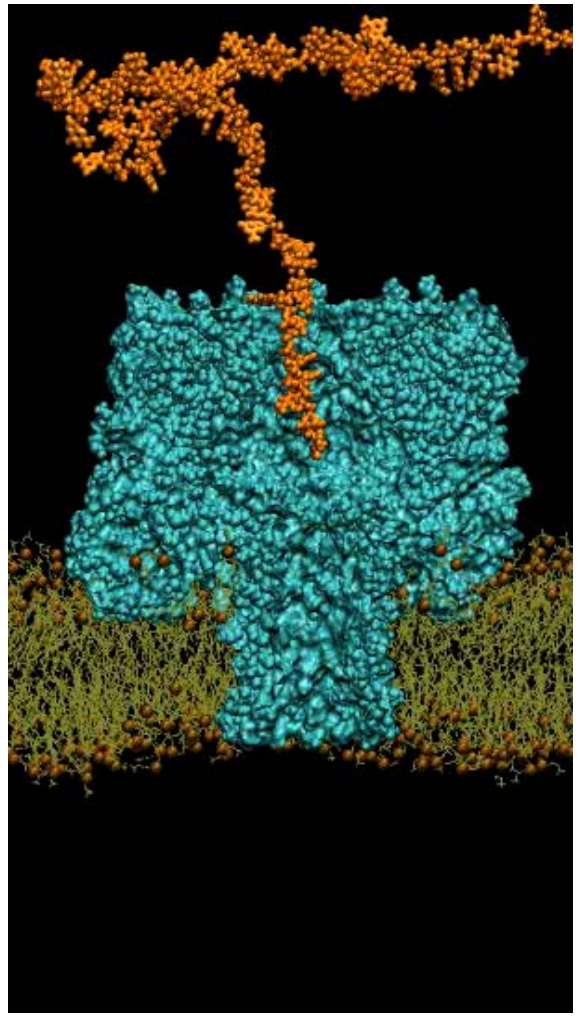
GPU Computing

- Commodity devices, omnipresent in modern computers
- Massively parallel hardware, hundreds of processing units, throughput oriented design:
- Programming tools allow software to be written in dialects of familiar C/C++ and integrated into legacy software
- 8x to 30x speedups common for data-parallel algorithms



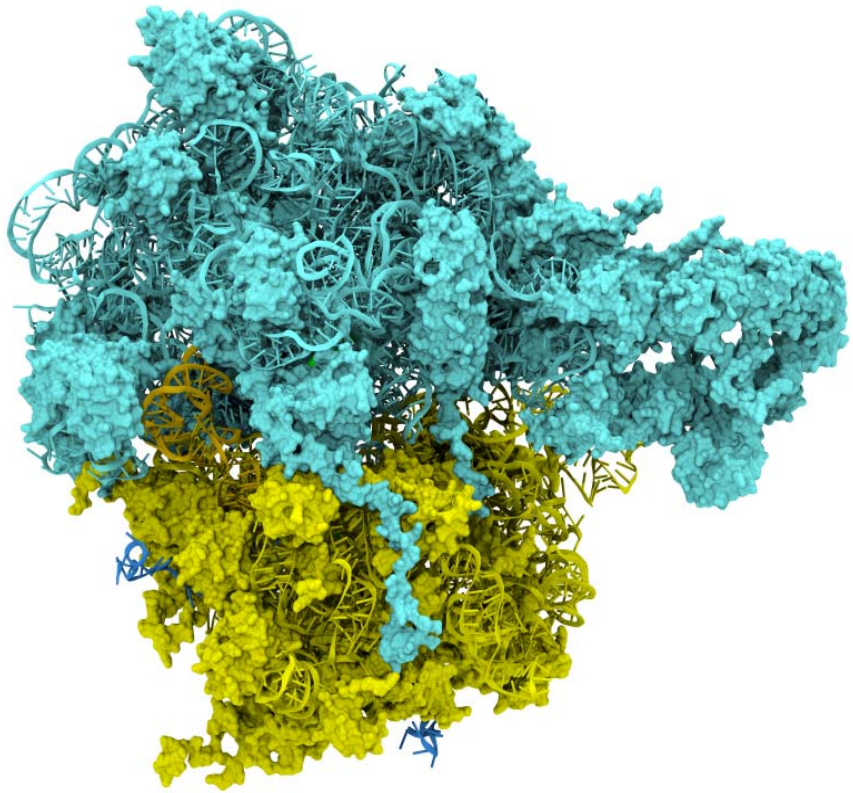
Computational Biology's Insatiable Demand for Processing Power

- Simulations still fall short of biological timescales
- Large simulations extremely difficult to prepare, analyze
- Order of magnitude increase in performance would allow use of more sophisticated models



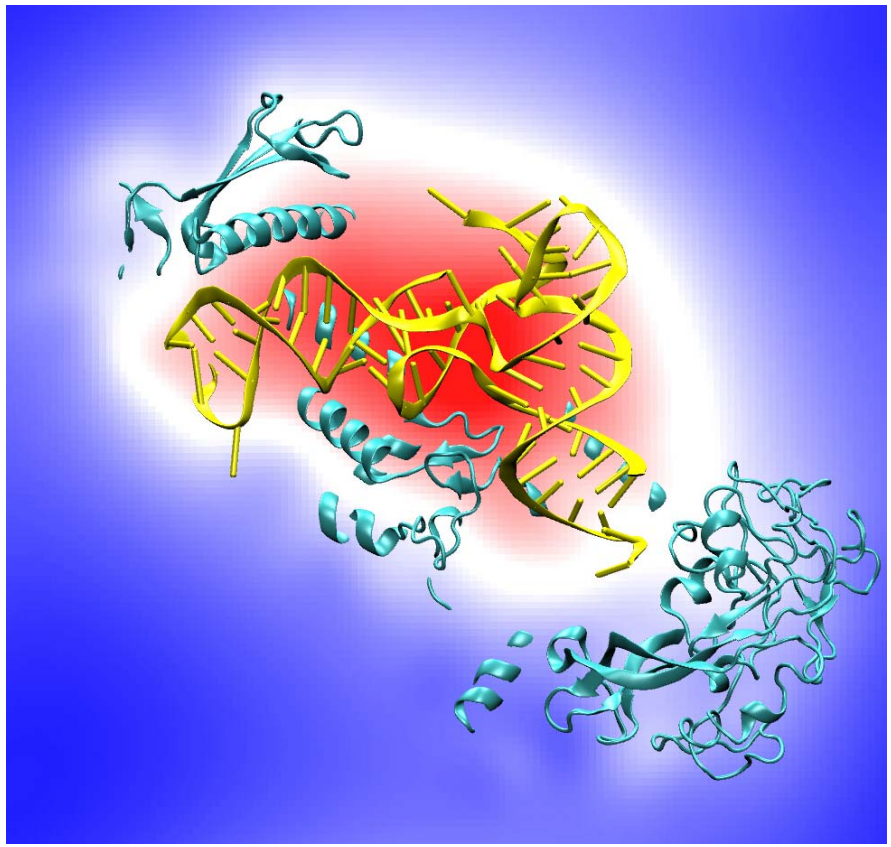
Accelerating Molecular Dynamics with Graphics Processors

- Jim Phillips will present GPU results for NAMD later today:
  - MS10 Atlanta A: 4:15-4:40pm



Calculating Electrostatic Potential Maps

- Used in structure building, analysis, visualization, simulation
- Electrostatic potentials evaluated on a uniformly spaced 3-D lattice
- Each lattice point contains sum of electrostatic contributions of all atoms



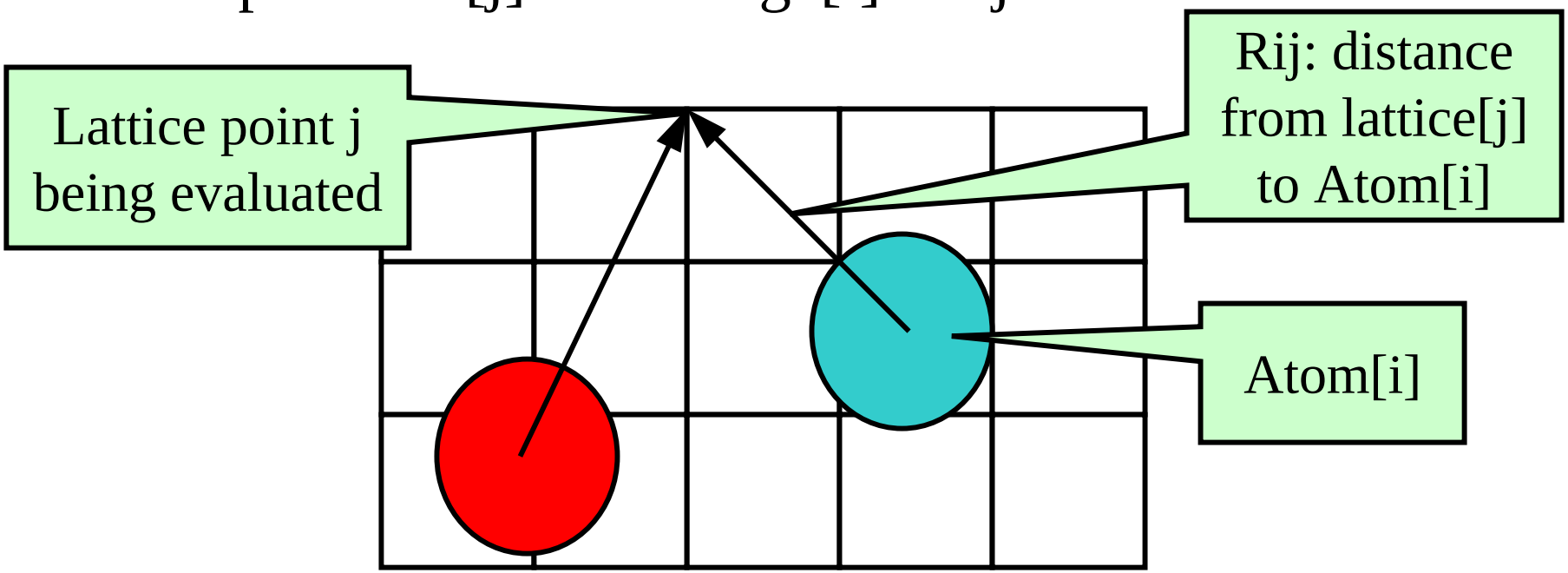
Positive potential field

Negative potential field

Direct Coulomb Summation

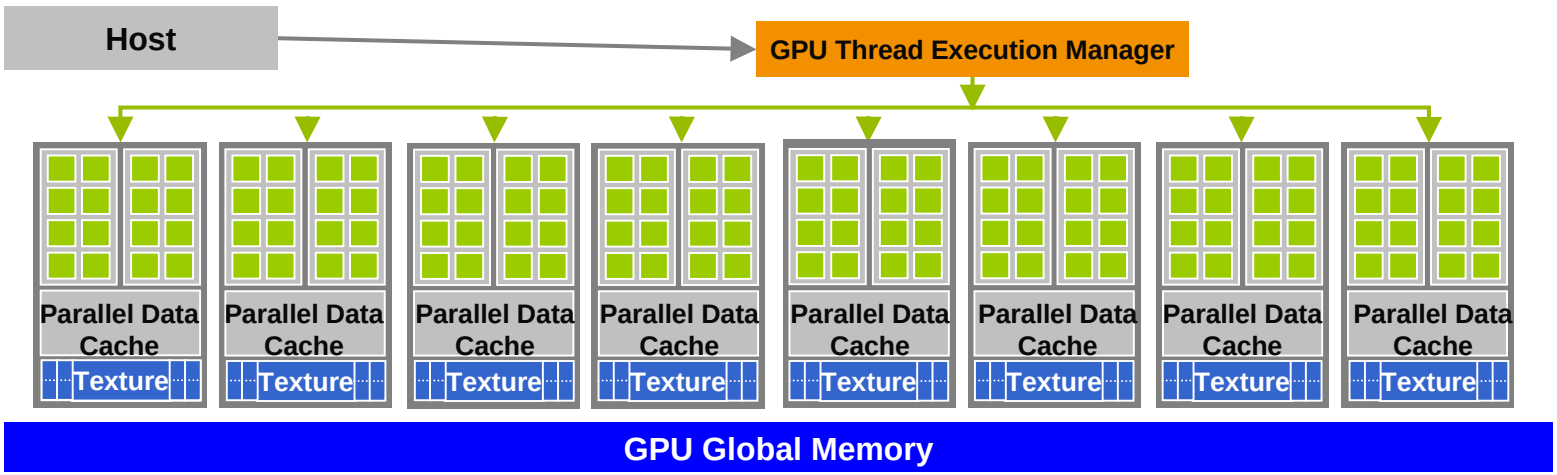
- At each lattice point, sum potential contributions for all atoms in the simulated structure:

potential[j] += charge[i] / Rij

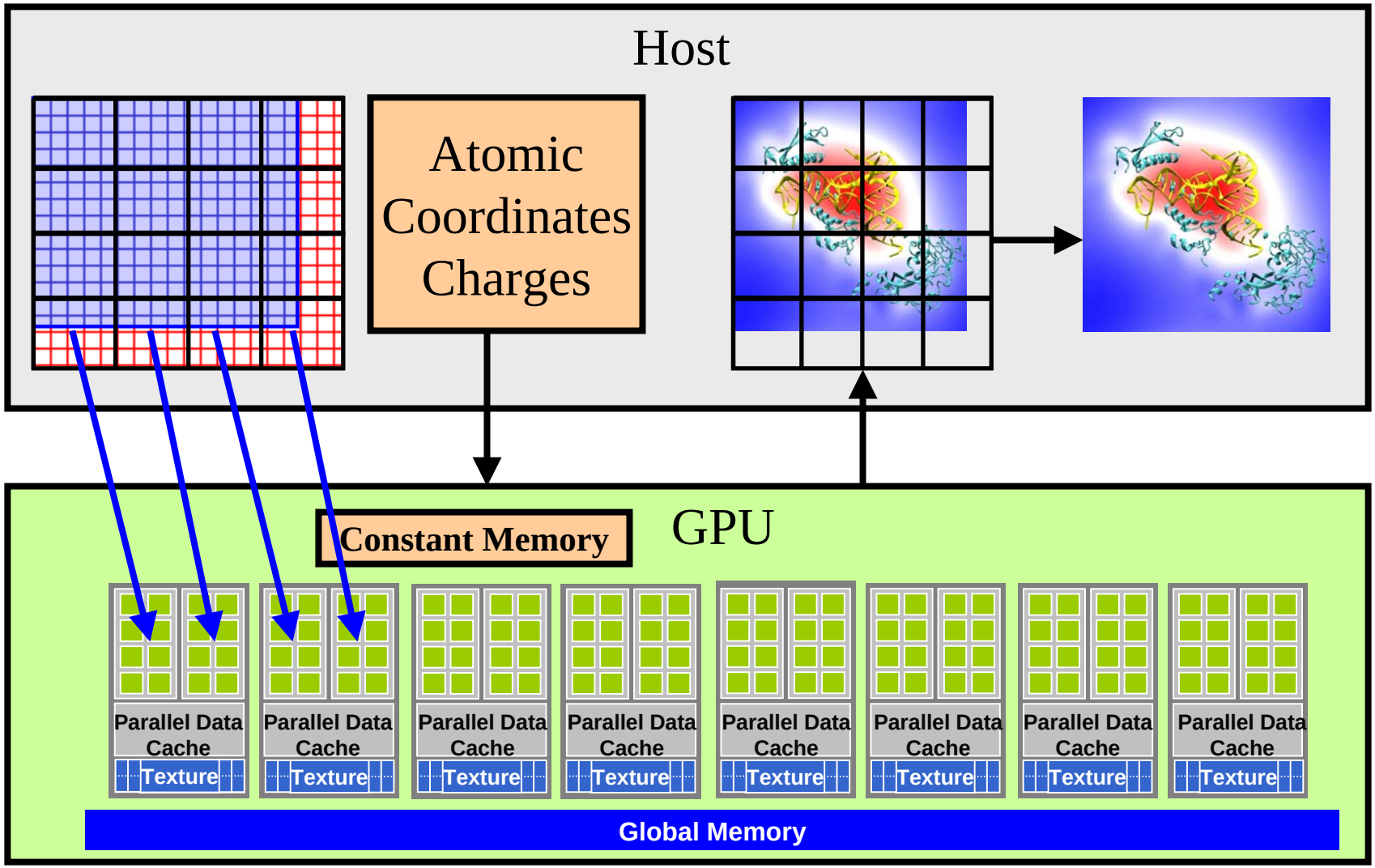


Direct Coulomb Summation on the GPU

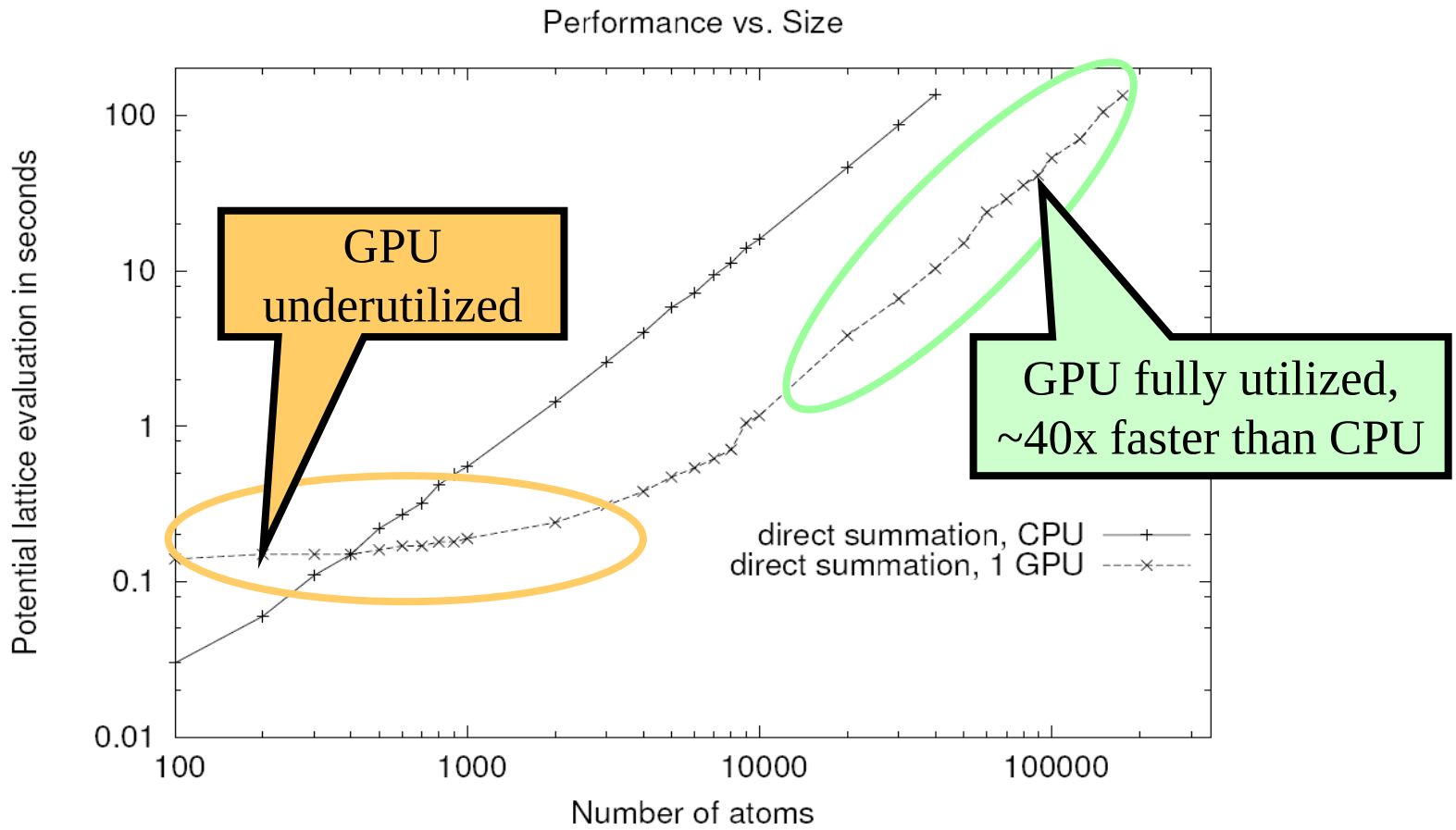
- GPU can outrun a CPU core by 44x
- Work is decomposed into tens of thousands of independent threads, multiplexed onto hundreds of GPU processor cores
- Single-precision FP arithmetic is adequate for intended application
- Numerical accuracy can be further improved by compensated summation, spatially ordered summation groupings, etc
- Starting point for more sophisticated algorithms



Direct Coulomb Summation on the GPU



Direct Coulomb Summation Runtime



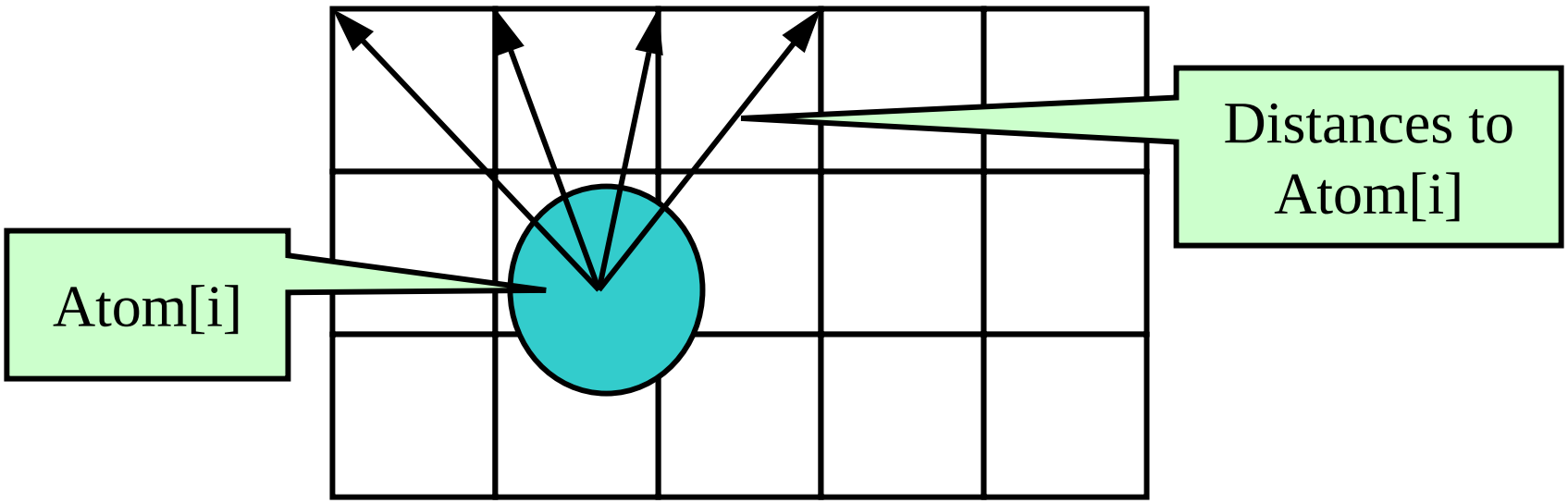
Accelerating molecular modeling applications with graphics processors.
J. Stone, J. Phillips, P. Freddolino, D. Hardy, L. Trabuco, K. Schulten.
J. Comp. Chem., 28:2618-2640, 2007.

Optimizing for the GPU

- Increase arithmetic intensity, reuse in-register data by "unrolling" lattice point computation into inner atom loop
- Each atom contributes to several lattice points, distances only differ in the X component:

  potentialA += charge[i] / (distanceA to atom[i])

  potentialB += charge[i] / (distanceB to atom[i]) …

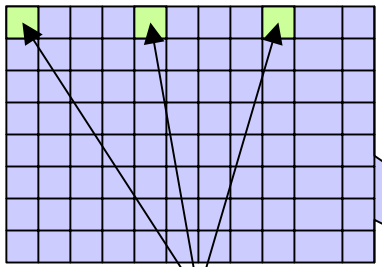


CUDA Block/Grid Decomposition

Unrolling increases computational tile size

Grid of thread blocks:

Thread blocks: 64-256 threads



Threads compute up to 8 potentials. Skipping by half-warps optimizes global mem. perf.

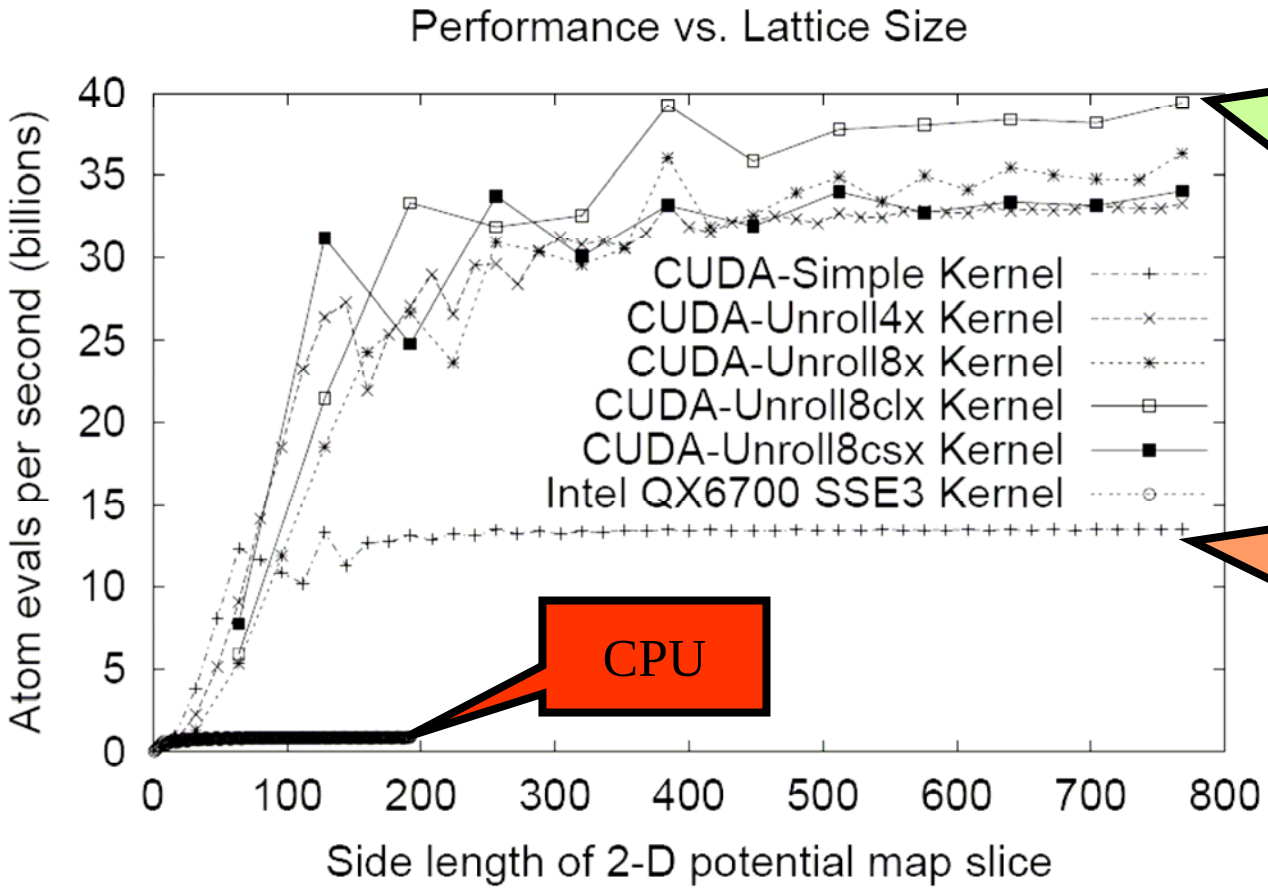
0,0 0,1 …

1,0 1,1 …

… … …

Padding waste

Direct Coulomb Summation Performance



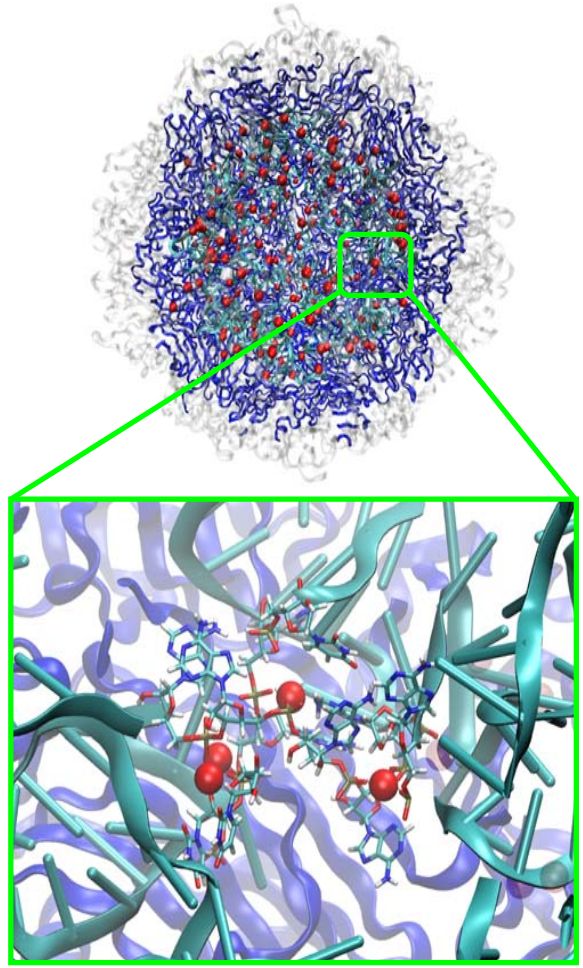
CUDA-Unroll8clx: fastest GPU kernel, 44x faster than CPU, 291 GFLOPS on GeForce 8800GTX

CUDA-Simple: 14.8x faster, 33% of fastest GPU kernel

GPU computing. J. Owens, M. Houston, D. Luebke, S. Green, J. Stone, J. Phillips. *Proceedings of the IEEE*, 2008. In press.

GPU Application Performance
(July 2007, current kernels are 20% faster...)

- CUDA ion placement lattice calculation performance:
  - 82 times faster for virus (STMV) structure
  - 110 times faster for ribosome
- Virus ion placement:
  110 CPU-hours on SGI Altix Itanium2
- Same calculation now takes 1.35 GPU-hours
- 27 minutes (wall clock) if three GPUs are used concurrently



Satellite Tobacco Mosaic Virus (STMV) Ion Placement

Multi-GPU Direct Coulomb Summation

- Effective memory bandwidth scales with the number of GPUs utilized
- PCIe bus bandwidth not a bottleneck for this algorithm
- 117 billion evals/sec
- 863 GFLOPS
- 131x speedup vs. CPU core
- Power: 700 watts during benchmark

Quad-core Intel QX6700

Three NVIDIA GeForce 8800GTX

Multi-GPU Direct Coulomb Summation

- 4-GPU (2 Quadroplex) Opteron node at NCSA
- 157 billion evals/sec
- 1.16 TFLOPS
- 176x speedup vs. Intel QX6700 CPU core w/ SSE

NCSA GPU Cluster

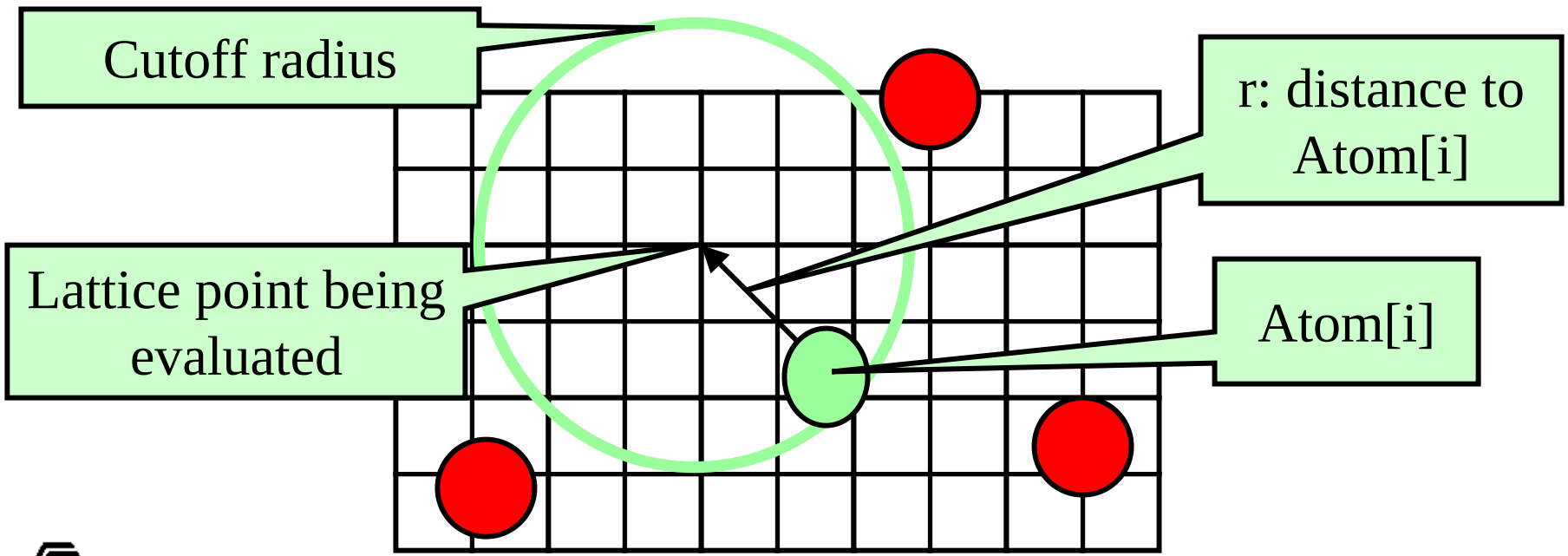
Cutoff Summation

- At each lattice point, sum potential contributions for atoms within cutoff radius:

  if (distance to atom[i] < cutoff)

    potential += (charge[i] / r) * s(r)

- Smoothing function s(r) is algorithm dependent



Infinite vs. Cutoff Potentials

- Infinite range potential:
  - All atoms contribute to all lattice points
  - Summation algorithm has quadratic complexity
- Cutoff (range-limited) potential:
  - Atoms contribute within cutoff distance to lattice points
  - Summation algorithm has linear time complexity
  - Has many applications in molecular modeling:
    - Replace electrostatic potential with shifted form
    - Short-range part for fast methods of approximating full electrostatics
    - Used for fast decaying interactions (e.g. Lennard-Jones, Buckingham)

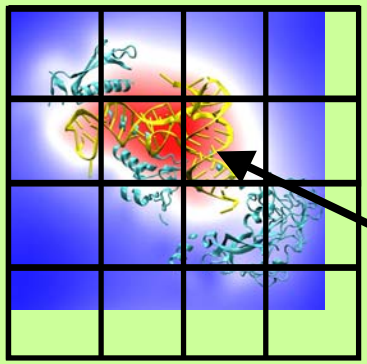
Cutoff Summation on the GPU

Atoms spatially hashed into fixed-size “bins” in global memory

Atoms

Global memory

Potential map regions

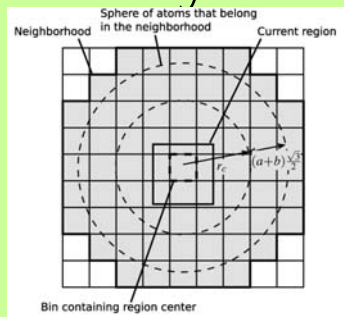


Shared memory

Atom bin

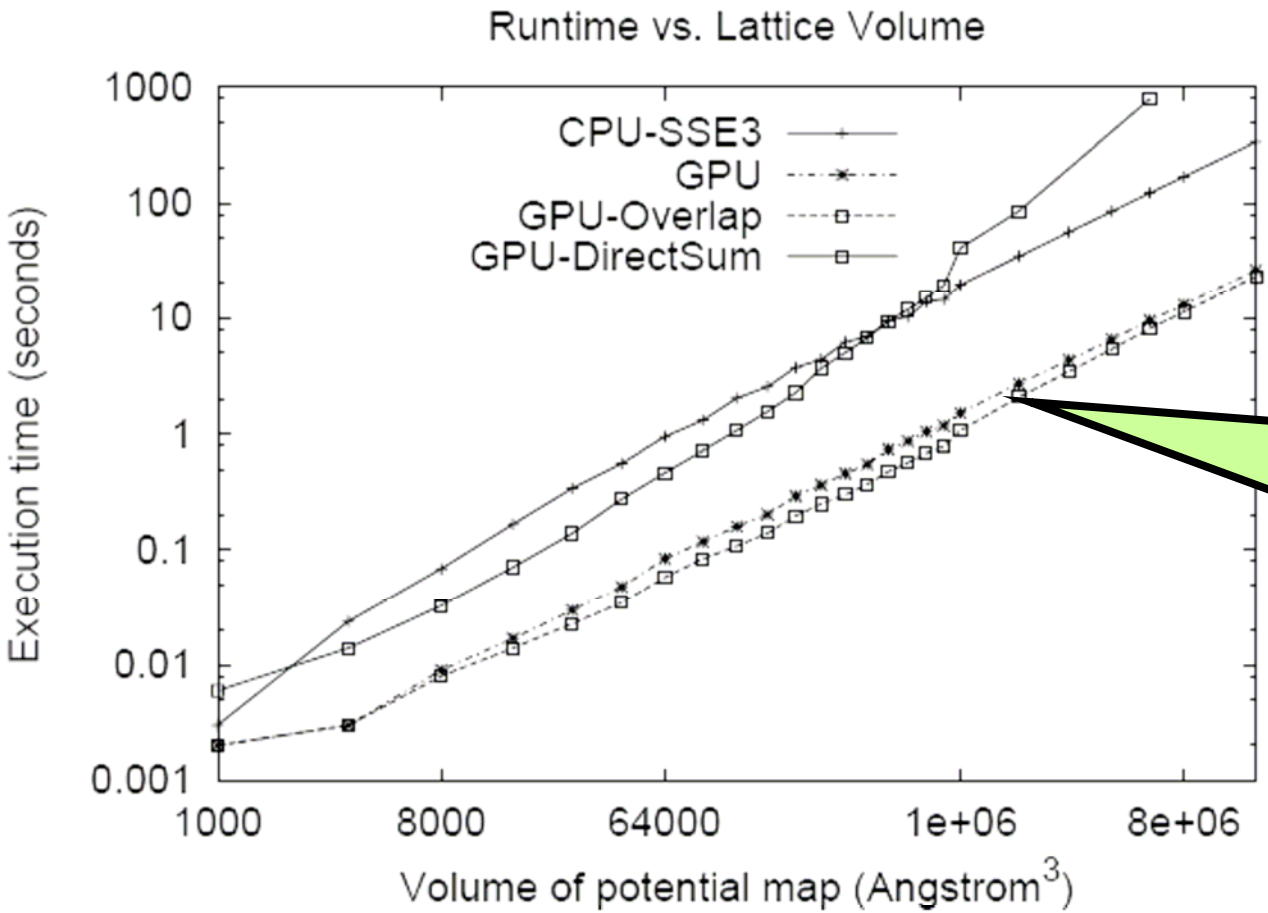
Constant memory

Bin-Region neighborlist



Process atom bins for current potential map region

Cutoff Summation Runtime



GPU cutoff with CPU overlap: 12x-21x faster than CPU core

GPU acceleration of cutoff pair potentials for molecular modeling applications. C. Rodrigues, D. Hardy, J. Stone, K. Schulten, W. Hwu. *Proceedings of the 2008 Conference On Computing Frontiers*, 2008. In press.

GPU Performance Results, March 2008

GeForce 8800GTX w/ CUDA 1.1, Driver 169.09

Calculation / Algorithm	Algorithm class	Speedup vs. Intel QX6700 CPU core
Fluorescence microphotolysis	Iterative matrix / stencil	12x
Pairlist calculation	Particle pair distance test	10-11x
Pairlist update	Particle pair distance test	5-15x
Molecular dynamics non-bonded force calculation	N-body cutoff force calculations	10x 20x (w/ pairlist)
Cutoff electron density sum	Particle-grid w/ cutoff	15-23x
Cutoff potential summation	Particle-grid w/ cutoff	12-21x
Direct Coulomb summation	Particle-grid	44x

http://www.ks.uiuc.edu/Research/gpu/

Lessons Learned

- GPU algorithms need fine-grained parallelism and sufficient work to fully utilize hardware
- Much of GPU algorithm optimization revolves around efficient use of multiple memory systems
- Amdahl's Law can prevent applications from achieving peak speedup with shallow GPU acceleration efforts

Acknowledgements

- Theoretical and Computational Biophysics Group, University of Illinois at Urbana-Champaign
- Prof. Wen-mei Hwu, Chris Rodrigues, IMPACT Group, University of Illinois at Urbana-Champaign
- David Kirk and the CUDA team at NVIDIA
- NIH support: P41-RR05969

Publications

- http://www.ks.uiuc.edu/Research/gpu/
- Accelerating molecular modeling applications with graphics processors. J. Stone, J. Phillips, P. Freddolino, D. Hardy, L. Trabuco, K. Schulten. *J. Comp. Chem.*, 28:2618-2640, 2007.
- Continuous fluorescence microphotolysis and correlation spectroscopy. A. Arkhipov, J. Hüve, M. Kahms, R. Peters, K. Schulten. *Biophysical Journal*, 93:4006-4017, 2007.
- GPU computing. J. Owens, M. Houston, D. Luebke, S. Green, J. Stone, J. Phillips. *Proceedings of the IEEE*, 2008. In press.
- GPU acceleration of cutoff pair potentials for molecular modeling applications. C. Rodrigues, D. Hardy, J. Stone, K. Schulten, W. Hwu. *Proceedings of the 2008 Conference On Computing Frontiers*, 2008. In press.